

Numere las hojas entregadas. Complete en la primera hoja la cantidad total de hojas entregadas. Cada ejercicio se califica con Bien, Regular o Mal. La división de los ejercicios en incisos es meramente orientativa. Los ejercicios se califican globalmente. El parcial se aprueba con 2 ejercicios Bien y a lo sumo 1 Mal. El parcial NO es a libro abierto; pero puede tener la Arquitectura de ORGA1 y dos hojas A4 con apuntes propios. **Importante:** Justifique sus respuestas. Las soluciones a ejercicios de la práctica que se utilicen deben ser incluidas en el examen.

Ejercicio 1 El sistema de representación de números fraccionarios de punto flotante llamado N-SPERA es un sistema de 8 bits con las siguientes características: los primeros 4 bits corresponden al exponente en complemento a 2, el siguiente bit al signo (del número) y los siguientes 3 a la mantisa del número.

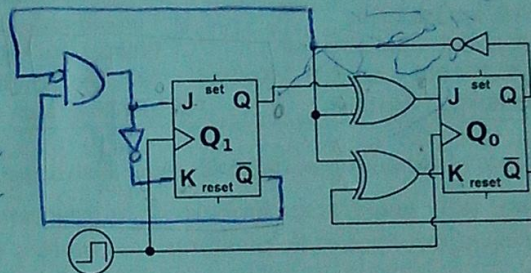
7	6	5	4	3	2	1	0
Exponente				Signo	Mantisa		

Esta representación es normalizada a "1," y se desnormaliza cuando el exponente toma el mínimo valor posible. En este último caso, el exponente vale el mínimo valor posible que no desnormaliza la representación.

- Indicar, en decimal, qué número representa la tira de bits 10011011 en este sistema de representación.
- ¿Cuál es el **intervalo de underflow**? (**IMPORTANTE:** El intervalo de underflow se define como $(-r_{min}, 0) \cup (0, r_{min})$ con r_{min} el mínimo real positivo representable).
- Dar un ejemplo de un número que genere overflow.
- ¿Es posible representar el número $0,004_{10}$ en este sistema? En caso afirmativo dar su representación. En caso negativo dar su representación lo más aproximada posible en este sistema.
- Indicar si es posible representar al cero con este sistema de representación. En caso negativo, justificarlo adecuadamente. En caso afirmativo mostrar su representación.
- ¿Existe en este sistema algún número que posea más de una representación posible? En caso afirmativo menciónelo y dé por lo menos dos representaciones de dicho número en este sistema.

Ejercicio 2 Un fabricante de hardware poco conocido nos otorgó el siguiente circuito, con su correspondiente tabla característica. Sin embargo, y para nuestra desgracia, ambos se encuentran incompletos.

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	0



- Completar la tabla y el circuito de manera que resulten consistentes entre sí.
- El circuito tiene un estado desde el cual, si se ejecuta por suficientes ciclos de clock, se llega a todos los demás. Llamaremos a este estado INICIAL. Decidir cuál es y realizar un diagrama de estados que refleje la sucesión de valores almacenados internamente por el circuito.
- Modificar el circuito realizado para que al llegar al estado $Q_1 = 1$ y $Q_0 = 1$ deje de cambiar de valor para siempre.

Ejercicio 3

- a) Dado el siguiente código escrito en lenguaje ensamblador de ORGA1 que se quiere cargar a partir de la posición de memoria 0xBEB0:

```

main:  DW 0xCAFE
       CMP R1, [0x1234]
       JGE fin
       SUB R2, [[0x0005]]
       JL main
sDos:  ADD R7, [R6 + 0x3529]
       JMP main
fin:   DW 0xCACA
    
```

I. Calcular el valor de las etiquetas

II. Indicar para cada instrucción que contenga una etiqueta por cuál valor reemplazará el ensamblador a dicha etiqueta. Expresar el valor en hexadecimal y en caso de ser un desplazamiento también en decimal.

III. Indicar el contenido de cada posición de memoria luego de ensamblado y cargado el código.

- b) Prendiendo una máquina ORGA1 cuyos valores iniciales son los siguientes:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0x6410	0x641B	0x43AC	0x500F	0xD9E4	0x5B01	0x8710	0x1800	0x6410	0x1870	0x1C40	0x0000	0x2800	0x0001	0xA000	0x641B	0x0000

Todo lo demás se encuentra en 0, excepto el SP que está en su valor por defecto y el PC que contiene el valor 0x6416.

I. Realizar el seguimiento hasta poder indicar si la ejecución termina o no. Realizar al menos 6 ciclos de instrucción.

II. En caso de que la máquina se detenga, indicar el valor final del PC y explicar detalladamente qué causó la finalización de la ejecución.

Ejercicio 4 Se nos contrata para diseñar el formato de instrucción de una nueva máquina controladora de una cadena de montaje de automóviles. La misma utilizará una memoria de 16 bytes direccionable a *byte*.

La cantidad de autos construidos de cada tipo estará guardada en una posición de memoria.

La cadena permite tomar un chasis de auto de 3, 4 ó 5 puertas; unos metros más adelante, colocarle la cantidad de puertas correctas; luego, colocarle un motor económico o uno turbo; y finalmente, unos metros más adelante, elegir entre 4 colores de pintura. Al terminar un auto, deberá incrementarse en uno el contador correspondiente a ese modelo.

La fábrica además tendrá instalados dos sensores por seguridad.

El set de instrucciones necesario se resume a:

Instrucción	Significado
TOMAR_CHASIS opcionChasis	Permite elegir entre chasis de 3, 4 ó 5 puertas.
PONER_PUERTAS opcionChasis	Indica a los brazos que debe poner 3, 4 ó 5 puertas.
MOVER_CINTA	Mueve la cinta transportadora un tiempo fijo.
PINTAR opcionColor	Pinta el auto con uno de los 4 colores.
PONER_MOTOR opcionMotor	Indica a los brazos que coloquen un motor económico o uno turbo.
SENSAR_CHASIS sensor, [dirMem]	Almacena en dirMem el valor de uno de los dos sensores.
SUMAR_MODELO [dirMem]	Incrementa en uno el contenido de dirMem.

- a) Indique el tamaño que deberán tener las direcciones de memoria de esta máquina.
- b) Se nos pide generar un formato de instrucción de 4 bits, exceptuando las instrucciones que contienen direcciones de memoria, que podrán tener más. ¿Es esto posible? Justifique.
- c) Indique si es posible agregar instrucciones sin operandos. En caso afirmativo, dar la cantidad.
- d) Sin modificar el set de instrucciones. ¿Es posible dar un mecanismo para proveer al programador de la siguiente funcionalidad?:

ARMAR_AUTO opcionColor, opcionChasis, opcionMotor, dirMem

Justifique.